Remarks

Claims 1-37 and 40-96 are currently pending in the application. Claims 38 and 39 have been cancelled. The Examiner has restricted the claims to one of two groups: Group I including Claims 1-62 and 82-96 and Group II including Claims 63-81. The Applicant's attorney made a provisional election of Group I during a phone conversation with the Examiner on January 25, 2006 and that provisional election is hereby affirmed. The Group II Claims 63-81 have been withdrawn from the application.

The Applicant has amended several of the pending claims in the application, however, these amendments were made in response to rejections based on informalities as noted in the next paragraph, and to improve the readability of the claims. No claim amendments were made in response to, or to distinguish the claims from, any of the prior art cited by the Examiner.

Claims 27-30 have been objected to because of informalities which have been remedied by amendments thereto. In addition, Claims 24-30 have been rejected under 35 U.S.C. § 112, second paragraph as being indefinite because those claims reference a method while the parent claim from which they depended reference a system. The deficiencies of those claims have been remedied by the amendments above as well. The Examiner states that Claims 38 and 39 have been rejected under 35 U.S.C. § 112, first paragraph as failing to comply with the enablement requirement. Those claims have been cancelled.

The Examiner has rejected claims 1-4, 7, 9-16, 53, 85-87, 89 and 92 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,609,132 (White, et al., hereinafter "White"). Claim 1 of the current application reads as follows:

26

1. A data management system in a computing environment comprising:

a. a data instance centric architecture;

b. where each data instance is encapsulated in a common fundamental data structure; and

c. where said common fundamental data structure also encapsulates references to associated separately encapsulated data instances.

The Applicant is of the opinion that none of the elements of Claim 1 are disclosed by White. One key aspect of the present invention is that each data instance encapsulates references to all other, associated data instances within a common, fundamental data structure in which it is encapsulated. This feature is expressed in element (c.) of Claim 1 and the Applicant is of the opinion that this limitation is not taught by White, and that, in fact, White teaches the direct opposite of this limitation, namely, that the relationships between data instances (or "data objects" as they are referred to in White) are kept separate from the data instances themselves.

Relationships between data objects in White are established by forming a predicate of the form "<subject object(s)> <modifier text> <direct object(s)>", where the subject object and the direct objects are data objects having some form of relationship or association with each other defined by a text string stored in <modifier text>. *See* White, column 5, lines 35-36. These relationship predicates, however, are not encapsulated within the data structures which encapsulate the data objects. Instead, they are kept as records in a database table, separate from the data objects. White states the following at column 6, lines 9-22:

> Preferably, the bi-directional modifier text for a given
> relation comprise arbitrary text strings defined by user input.
> Moreover, *the data structures (i.e., data records) representing the*

> *given relation is preferably separate from (and indirectly coupled
> to) the data structures representing the subject object(s) and
> direct object(s)*; thus, in this case, the bi-directional modifier text is
> not defined by (and thus can differ from) any fields (attributes) of
> the data structures that represent the subject object(s) and direct
> object(s) of the given relation. (emphasis added)

The italicized portion of the passage above clearly shows that the relationship

predicates (i.e., "the data structures representing the given relation") are kept separate from

the data objects (i.e., "the data structures representing the subject objects"). This is the direct

opposite of the present invention, wherein an association between a first data object and a

second data object is indicated by the encapsulation of a reference to the second data object

within the fundamental data structure encapsulating the first data object, and visa versa. Also,

in the present application, groupings of the encapsulated references is also indicative of the

type of association between the referenced data objects (*See* claim 7).

In addition, the Applicant respectfully submits that the implementation of the White

system does not represent a data instance centric architecture. In a data instance centric

architecture, the data instances, encapsulated in common fundamental data structures, make

up the entirety of the database. No tables are needed in which data objects, relationship

predicates, relation types, or data types are stored. In addition, the data instance centric

architecture also means that each data instance is encapsulated in a common fundamental data

structure. White, however, represents a "table centric architecture" wherein both the data

objects as well as relations and the relation types are stored in simple relational database

tables. The Applicant directs the Examiner's attention to White at column 7, line 62 through

column 8, line 3, which states as follows:

> The logical data structures of the illustrative embodiment of
> FIGS. 2 and 3 *may be embodied in any arbitrary database model*
> including a relational database model, object database model,
> object-relational database model, network database model, or

> hierarchical database model. FIG. 4 illustrates an embodiment of
> the logical data structures of FIGS. 2 and 3 in a relational database
> model, including an Object Table, Relation Table, Relation Object
> Table, Modifier Table, Relation Type Table, and Object Type
> Table. (emphasis added)

Therefore, in the White application, some form of database structure is required to implement the system, while in the present application, no database structure is required. The database itself consists of the entire collection of common fundamental data structures encapsulating the data instances along with their relations. As a result, the Applicant respectfully submits that this the White system is not a "data instance centric architecture" (element (a.) of Claim 1), nor does the White system have all data instances encapsulated in common fundamental data structures (element (b.) of Claim 1).

The Applicants respectfully submit that none of elements (a.), (b.) or (c.) of Claim 1 have been met by White, and that, in fact White teaches the direct opposite of all of these elements. As a result, the Applicants respectfully submit that Claim 1 and all of its dependent claims are patentably distinct from White and that the rejection under 102(e) based on White has been traversed.

With respect to Claim 2, the Examiner states that the feature of structural symmetry is met by White at column 5, lines 48-63 and column 7, lines 18-38. However, this section of White discusses bi-directional modifiers which define the relation between a subject object and a direct object, and notes that the White system requires a second text string that characterizes the semantics of the relationship of the direct object to the subject object of the given relation. *See* White at column 5, lines 53-55. This feature of White destroys the structural symmetry of the system. In the system of the present application, a relationship between a first data instance and any other data instance is indicated by the presence of a reference to the other data instances in the encapsulation of the first data instance. The type of

association is indicated by the groupings of the references within the encapsulation of the first data object, the groupings being indicated by a multi-dimensional logical index (i.e., the encapsulated reference). There is no second reference that indicates the opposite relation. The opposite relation is indicated by the mutual inclusion of the references in the encapsulation of each data object.

With respect to Claim 3, the Examiner states that the feature that data instances are encapsulated with references to associated data instances is disclosed in White at column 6, lines 22-43 and column 7, lines 18-38. However, this passage of White discusses the types of relations, namely the object relations and type relations, both being represented by a data structure that stores textual annotation characterizing the semantics of the relationship. However, as shown previously, these data structures are not encapsulated with the data instances themselves. As discussed with respect to Claim 1, they are preferably stored separately and away from the related data objects (i.e., subject objects and direct objects in White). Thus, the same discussion with respect to Claim 1 applies here; the only difference being that Claim 3 adds the limitation that associated data instances include mutual references to each other within their encapsulations.

With respect to Claim 7, which includes the limitation that each of the dimensions in the multi-dimensional encapsulated reference corresponds to a type of association, the Examiner states that this is disclosed in White at column 7, lines 5-11. However, this passage in White merely states that there are references which identify objects that belong to a given object type. No mention is made in this passage or anywhere else in White regarding the type of association. The multi-dimensional references of the present application provide a convenient way to indicate that several data objects are of the same type. For instance, three

data instances may have logical indices as follows: <X, Y, Z, 1>, <X, Y, Z, 2>, <X, Y, Z, 3>. This indicates that each of the data instances are the first, second and third examples of data instances having a particular "type" indicated by the logical reference <X, Y, Z>

With respect to Claim 89, which adds the limitation that references to associated items are arranged in sets, defining the type of association between the items and each of the items in the referenced set, the Examiner states that this is disclosed in White at column 7, lines 44-61, which discusses a relation type table entry. First of all, referring back to Claim 85, Claim 85 includes the limitation of Claim 1 wherein items which are associated with each other encapsulate mutual references to each other. As stated with respect to Claim 1, the data objects in White do not encapsulate references to associated items but instead are related to other items via entries in a relationship table. Therefore, with respect to Claim 85, the limitation of mutual encapsulated references is not met by White. With respect to claim 89, which depends from Claim 85, Claim 89 adds the limitation that within an encapsulated data instance, the references to other items are arranged in sets which define the type of association between the encapsulating item and the encapsulated references. The cited passage in White, however, discloses the relation type table entry which is an entry in a relational database which defines the type of relationships between a group of objects and another single object. However, the groups of objects are not arranged in encapsulated sets of references which define the association but are referenced by table entries from a separate table entry which defines the relationship between the single data object and the set of other dated objects. In addition, the objects referenced by the subject type ID's in relation type table entry do not contain mutual references back to the original data object type. Therefore, the limitations of Claim 89 are not met.

With respect to Claim 92, which states that items may act as containers for other member items, the Examiner states that this is disclosed in White at column 6, lines 66 through column 7, line 11. This passage of White discloses entries in a relational database table that identified types of objects. For each object, there is a table entry in a relational database table which includes the type and one or more object identifiers showing objects of that type. This is not the same as an encapsulated data instance of the present application wherein the data instance may act as a container for other member items. In the present application, the contained data is actually included in the encapsulation that includes that data element itself. This is not disclosed by White.

The Examiner has rejected Claims 82 through 84 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,711,582 (Aldridge, et al., hereinafter "Aldridge"). These claims include a method for converting a table-based database system into the data instance centric database system of the present application. Aldridge, however, discloses a method for mapping an object-oriented object to one or more relational database tables. *See* Aldridge at Abstract, column 1, lines 15-18 and column 9, lines 10-25 wherein it is described that the method of Aldridge is useful for mapping data between an object oriented data model and a relational data model. Note that the object oriented data model more closely resembles the data instance centric architecture of the present application, while the relational database tables are an example of a non-data instance centric architecture referred to in the present application. The method of Claims 82-85 are useful for mapping between a relational database and a data instance centric database of the present application whereas Aldridge discloses the exact opposite, *i.e.*, mapping data in an object oriented data model to a relational

database model. As a result, the Applicant respectfully disagrees with the Examiner in that Aldridge discloses the methods of Claims 82-85.

The Examiner has rejected Claims 5, 6, 8, 18, 19-24, 31-34, 36, 47-48, 50-52, 54-55, 58-60, 62, 88, 90 and 93 under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 5,809,297 (Kroenke, et al., hereinafter "Kroenke"). The Examiner states that White teaches the data management system of Claim 1, but does not explicitly recite that the logical index is 'm' dimensional in that it has 'n' bits per dimension as is recited in Claim 5. The Applicant respectfully submits that White teaches away from the present application as previously discussed with respect to Claim 1. As a result, there is no motivation to combine White with Kroenke. Further, Kroenke teaches a computer based system for allowing a user to create a relational database schema. This also teaches away from the present application. The present application discloses a data instance centric database as opposed to a non-data instance centric database, an example of which would be a relational database table. The Examiner states that Kroenke teaches a object data model for semantic relationships wherein the logical indices are 'm' dimensional and have 'n' bits per dimension. However, this is not what is disclosed in the cited passage in Kroenke. Instead, Kroenke describes a subscript for an attribute consisting of a pair of integers having the form 'm.n' where 'n' refers to the minimum cardinality and 'm' refers to the maximum cardinality of the attribute, where a minimum cardinality is a minimum number of instances of the attribute in the data record and the maximum cardinality is a maximum number of instances of the data record. This is not the same as the 'm' dimensional index referred to in Claim 5. Therefore, for several reasons, the combination of White and Kroenke does not teach or suggest the present application. First, both White and Kroenke teach away from the present

application. White teaches away in a sense that data objects and the relations from data objects are kept separately from each other, while Kroenke teaches away from the present application in that the purpose of the invention in Kroenke is to create a relational database schema which are part of a non-data instant centric architecture. Second, the Examiner's interpretation of Kroenke appears to be incorrect in that it does not disclose an 'm' dimensional index but discloses a one dimensional attribute having a minimum number of instances and a maximum number of instances. Therefore, there is no motivation or suggestion in either application for combining them to form the claimed invention. The remainder of the claims are rejected based on a combination of White and Kroenke all appear to include the argument with respect to Claim 5 for which the Applicant has just provided a rebuttal. Therefore the Applicant respectfully submits that this argument also applies to all other claims rejected based on a combination of White and Kroenke.

The Examiner has rejected Claim 27 under 35 U.S.C. § 103(a) as being unpatentable over White in view of Kroenke and further in view of U.S. Patent Application Publication No. 2003/0216169 (Walker, et al., hereinafter "Walker"). Walker is utilized for the disclosure of basic mathematical operations which result in the direct exclusion of one or more encapsulated references from the result of a comparison in a single operation. Walker appears to be utilized for the use of Boolean operations which further comprise basic mathematical operations. The Applicant concedes that Boolean operations are basic mathematical applications well known to those of skill in the art. However, as discussed above, with respect to the rejection based on the combination of White and Kroenke, both White and Kroenke teach away from the application and contain no motivation for combining themselves in the manner discussed by the Examiner, and further do not teach what the

Examiner has stated in the rejection. Therefore, with respect to Claim 27, a combination of White, Kroenke and Walker does not teach or suggest the claimed invention. These arguments apply also to Claims 28-30.

Claim 40 is rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 5,873,049 (Bielak, et al., hereinafter "Bielak"). The Examiner states that White does not explicitly disclose the system of Claim 1 further comprising encapsulated data instances representing ASCII characters but that Bielak teaches a database where ASCII characters are encapsulated in data objects. This portion of Bielak, however, teaches the storage of seismic data in an ASCII format. It does not explicitly teach wherein individual ASCII characters are encapsulated in separate data objects. In addition, as discussed previously, White teaches away from the present application in that it suggests a separate storage of data objects and relationships between data objects. Therefore, the combination of White and Bielak does not teach or discuss the invention claimed in claim 40, nor is there a suggestion for their combination.

The Examiner states that Claim 40 is rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent Application Publication No. 2003/0076978 (Eversole, et al., hereinafter "Eversole"). The Examiner states that White as applied to Claim 1 does not explicitly disclose encapsulated data instances representing Unicode characters. However, this section of Eversole merely suggests that an object property type can be identified as a Unicode string, not that Unicode characters can be encapsulated in common fundamental data structures of the present application. As a result, in addition as discussed above, White teaches away from the present application and therefore the combination of White and Eversole is both improper and does not suggest the invention claimed in Claim 42.

The Examiner has rejected Claim 44 as being unpatentable under 35 U.S.C. § 103(a) over White in view of U.S. Patent No. 5,812,840 ("Shwartz"). The Examiner states that White does not explicitly disclose the system of Claim 1 further comprising encapsulated data instances representing a token set of any data type. The Examiner states that Shwartz teaches a method and system for database wherein a set of encapsulated variables are included in an object data structure. However, Shwartz appears to teach a system wherein a database query assistant assists the user in constructing syntactically and semantically valid SQL statements. Claim 44 merely expands on Claim 1 to add the limitations that the encapsulated references to other data instances may be grouped into token sets containing references of any given type. This is not what is disclosed by Shwartz. Shwartz has no concept of encapsulation wherein an encapsulation entails encapsulating individual data instances along with all the information necessary to establish relationships between all other encapsulated data instances. Therefore, the combination of White and Shwartz does not disclose or teach the invention of Claim 44.

The Examiner has rejected Claims 17, 49 and 61 under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 6,957,214 (Silberberg, et al., hereinafter "Silberberg"). The Examiner says that with respect to Claim 17, White does not explicitly teach that one of the encapsulated references is a reference to an encapsulated data instance in another computing environment, but that Silberberg teaches an architecture for distributed database information access wherein data instances are located in different computing environments. The Applicant concedes that Silberberg teaches accessing data in a plurality of domains which are distributed. However, Silberberg fails to teach that the references are encapsulated within another encapsulated data instance that is associated therewith. White

36

also does not teach this feature as discussed previously. White teaches storing the relationships between objects in a table separate. Therefore, the combination of White and Silberberg does not teach that at least one encapsulated reference is a reference to an encapsulated data instance in another computing environment because White does not disclose the use of encapsulated references. The same argument applies to Claims 49 and 61.

Claims 94-96 are rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 6,016,497 ("Suver"). The Examiner states that White does not explicitly teach that items may encapsulate embedded items but that Suver teaches a method and system for storing and accessing embedded information in object relational databases wherein data instances encapsulate embedded elements. Suver, however, teaches the embedding of subtables within a relational database system, not wherein each data instance is encapsulated in the common fundamental data structure which can contain embedded elements. Furthermore, White, as discussed previously, teaches away from the present application. Therefore, the Applicant respectfully submits that the combination of White and Suver is improper, lacks motivation, and in combination does not teach the claimed invention.

The Applicant appreciates the Examiner's acknowledgement that Claims 25-26, 29-30, 35, 37, 41, 43, 45, 46 and 91 would be allowed if rewritten in independent form. However, the Applicant declines to do so at this time pending the Examiner's response to the arguments presented herein. The Applicant appreciates the Examiner's effort and the considerable time invested in the review of the invention and the thoughtful application of the cited references to the present application.
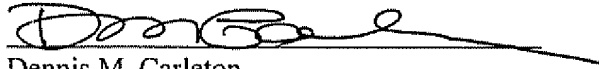
Conclusion

The Applicant has put forth reasoned arguments in support of the patentability of the rejected claims. With respect to White, the Applicant would like to point out that White is based on a relational database model and implemented using tables and as such, is not a data instance centric architecture as described in the present application. Further, White teaches away from the present application in that data instances are not fully encapsulated because the relationships with other data instances are captured in a table entry in a table which is separate from the data instance. Therefore, the rejection of claims based directly on White is traversed and the rejection of claims based on White and a combination of any other art is also traversed based on the fact that White teaches away from the application and therefore contains no suggestion to combine with any of the other cited references. As a result, the Applicant requests that the rejection of the rejected claims be withdrawn and a Notice of Allowance issued at the earliest possible time.

The Applicant believes that no additional fee is required with this Response. However, if any other fees are required for any reason, the Commissioner is hereby authorized to charge Deposit Account No. 02-4800 the necessary amount.

Should any issues remain, the Examiner is invited to contact the undersigned at the

number listed below to advance prosecution of the case.

Respectfully submitted,

Dennis M. Carleton
Registration No. 40,938

BUCHANAN INGERSOLL PC
(including the attorneys from Burns, Doane,
     Swecker & Mathis)
20th Floor, One Oxford Centre
301 Grant Street
Pittsburgh, Pennsylvania 15219-1410
Phone: 412-562-1895
Fax: 412-562-1041
e-mail: carletondm@bipc.com
Attorneys for Applicant(s)